NASA-CR-192990

LC .UI
GRANT
IN-39-CR
DATE OVERRIDE
44 P.
159797

PROMISS USER MANUAL

Prepared by:

Lola Boyce, Ph.D., P.E.
Thomas B. Lovelace

APPENDIX 1
of Final Technical Report
of Project Entitled
Development of Advanced Methodologies
for Probabilistic Constitutive Relationships
of Material Strength Degradation Models, Phase 2

NASA Grant No. NAG 3-867, Supp. 2

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Cleveland, OH 44135

The Division of Engineering
The University of Texas at San Antonio
San Antonio, TX 78285
January, 1990

Vers. 2.0

PROMISS USER MANUAL

Prepared by:

Lola Boyce, Ph.D., P.E.
Thomas B. Lovelace

APPENDIX 1
of Final Technical Report
of Project Entitled
Development of Advanced Methodologies
for Probabilistic Constitutive Relationships
of Material Strength Degradation Models, Phase 2

NASA Grant No. NAG 3-867, Supp. 2

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Cleveland, OH 44135

The Division of Engineering
The University of Texas at San Antonio
San Antonio, TX 78285
January, 1990

Vers. 2.0

TABLE OF CONTENTS

1.0 INTRODUCTION

This User Manual documents the FORTRAN program PROMISS (Vers. 2.0). The program determines the random strength of an aerospace propulsion component, due to a number of diverse random effects (see Section 2.0, Theoretical Background).

Included in this User Manual are details regarding the theoretical background of PROMISS, input data instructions and sample problems illustrating the use of both PROMISS and PROMISC. APPENDIX A gives information on the primitive variables, their symbols, FORTRAN names and both SI and U.S. Customary units. APPENDIX B includes a disk containing the actual input and output files corresponding to the sample problems. The source code is available from the first author at the address given on the cover page of this report. APPENDIX C details the IMSL, Version 10 [1], subroutines and functions called by PROMISS. APPENDIX D illustrates SAS/GRAPH [2] programs that can be used to plot both the probability density functions (p.d.f.) and the cumulative distribution functions (c.d.f.).

2.0 THEORETICAL BACKGROUND

Recently, a general phenomenological constitutive relationship, for composite materials subjected to a number of diverse effects or primitive variables, has been postulated to predict mechanical and thermal material properties [3,4,5,6]. The resulting multifactor interaction constitutive equations summarize composite micromechanics theory and have been used to predict material properties for a unidirectional fiber-reinforced lamina, based on the corresponding properties of the constituent materials.

These equations have been modified to predict the mechanical property of strength for one constituent material due to "n" diverse effects or primitive variables. These effects could include both time-independent and time-integrated primitive variables, such as mechanical stresses subjected to both static and impact loads, thermal stresses due to temperature variations and thermal shock, and other effects such as chemical reaction or radiation attack. They might also include other time-dependent primitive variables such as creep, mechanical fatigue, thermal aging, thermal fatigue, or even effects such as seasonal attack (see APPENDIX A, Primitive Variables, Symbols, and Units). For most of these primitive variables, strength has been observed to decrease with an increase in the variable.

The postulated constitutive equation accounts for the degradation of strength due to these primitive variables. The general form of the equation is

$$\frac{S}{S_O} = \prod_{i=1}^{n} \left[\frac{A_{iF} - A_i}{A_{iF} - A_{iO}} \right]^{a_i}, \qquad (1)$$

where A_i, A_{iF} and A_{iO} are the current, ultimate and reference values of a particular effect, a_i is the value of an empirical constant for the i^{th} effect or primitive variable, n is the number of product terms of primitive variables in the model, and S and S_O are the current and reference values of material strength. Each term has the property that if the current value equals the ultimate value, the current strength will be zero. Also, if the current value equals the reference value, the term equals one and strength is not affected by that variable.

This deterministic constitutive model may be calibrated by an appropriately curve-fitted least squares multiple linear regression of experimental data [7], perhaps supplemented by expert opinion. Ideally, experimental data giving the relationship between effects and strength is obtained. For example, data for just one effect could be plotted on log-log paper. A good fit for the data is then obtained by a linear regression analysis. This is illustrated schematically in Figure 1. The postulated constitutive equation, for a single effect, is then obtained by noting the linear relation between log S and $\log \left[\frac{A_F - A_O}{A_F - A} \right]$,

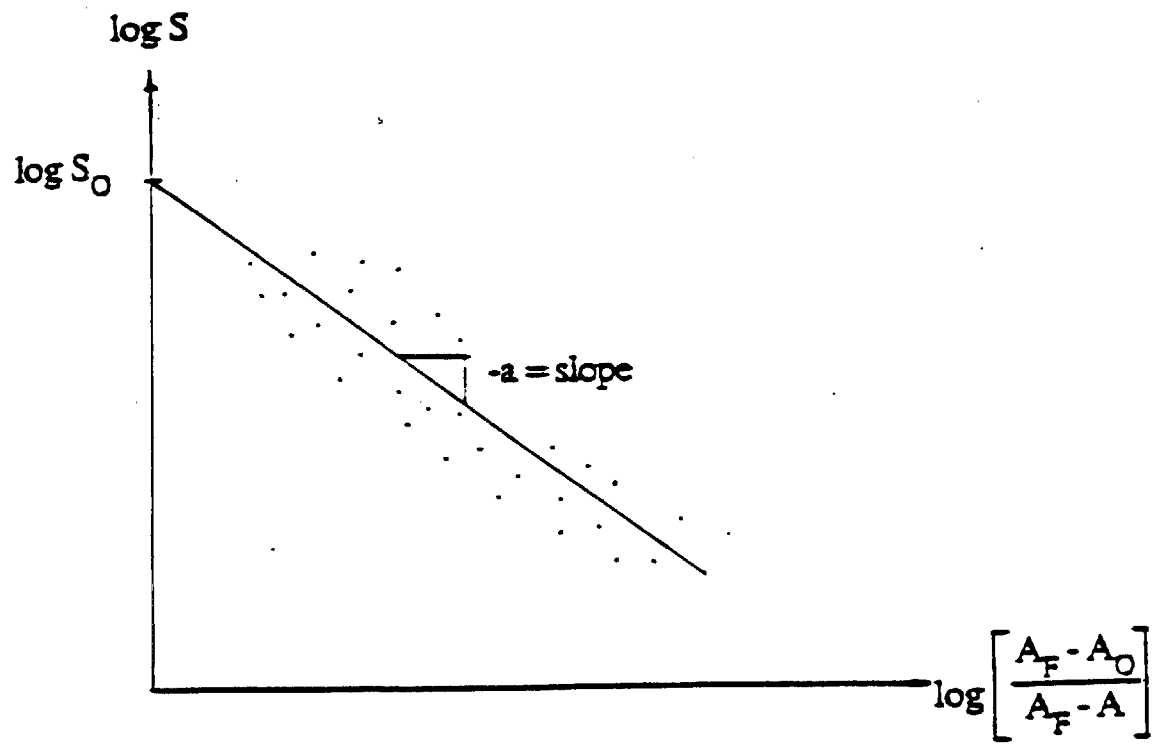

Fig. 1 Schematic of experimental data illustrating the effect of one primitive variable on strength.

as follows:

$$\log S = - a \log \left[\frac{A_F - A_O}{A_F - A} \right] + \log S_O$$

$$\log S - \log S_O = - a \log \left[\frac{A_F - A_O}{A_F - A} \right]$$

$$\log \frac{S}{S_O} = - a \log \left[\frac{A_F - A_O}{A_F - A} \right]$$

$$\frac{S}{S_O} = \left[\frac{A_F - A_O}{A_F - A} \right]^{-a}$$

$$\frac{S}{S_O} = \left[\frac{A_F - A}{A_F - A_O} \right]^{a} . \qquad (2)$$

Note that the above equation (2) is for a primitive variable that lowers strength. If a variable raises strength, the exponent is negative.

This general constitutive model may be used to estimate the strength of an aerospace propulsion system component under the influence of a number of diverse effects or primitive variables. The probabilistic treatment of this equation includes randomizing the deterministic multifactor interaction constitutive equation, performing probabilistic analysis by simulation and generating probability density function (p.d.f.) estimates for strength using a non-parametric method, maximum penalized likelihood [8,9]. Integration yields the cumulative distribution function (c.d.f.) from which probability statements regarding strength may be made. This probabilistic constitutive model predicts the random strength of an aerospace propulsion component due to a number of diverse random effects.

This probabilistic constitutive model is embodied in two FORTRAN programs, PROMISS (Probabilistic Material Strength Simulator) and PROMISC (Probabilistic Material Strength Calibrator); see Final Technical Report, APPENDIX 2. PROMISS calculates the random strength of an aerospace propulsion component due to as many as eighteen diverse random effects. Results are presented in the form of probability density functions and cumulative distribution functions of normalized strength, S/S_O. PROMISC calculates the values of the empirical material constants, a_i.

PROMISS includes a relatively simple "fixed" model as well as a "flexible" model. The fixed model postulates a probabilistic constitutive equation that considers the primitive variables given in Table 1. The general form of this constitutive equation is given in equation (1), wherein there are now n = 7 product terms, one for each effect or primitive variable listed above. Note that since this model has seven primitive variables, each containing four values of the variable, it has a total of twenty-eight variables. The flexible model postulates a probabilistic constitutive equation that considers up to as many as n = 18 product terms for primitive variables. These variables may be selected to utilize the theory and experimental data currently available for the specific strength degradation mechanisms of interest. The specific effects included in the flexible model are listed in Table 2. Note that in order to provide for future expansion and customization of the flexible model, six "other" effects have been provided.

Table 1 Primitive variables available in the fixed model

i^{th} Primitive Variable	Primitive Variable Type
1	Stress due to static load
2	Temperature
3	Chemical reaction
4	Stress due to impact
5	Mechanical fatigue
6	Thermal fatigue
7	Creep

Table 2 Primitive variables available in the flexible model

A. Environmental Effects

1. Mechanical
   a. Stress
   b. Impact
   c. Other Mechanical Effect

2. Thermal
   a. Temperature Variation
   b. Thermal Shock
   c. Other Thermal Effect

3. Other Environmental Effects
   a. Chemical Reaction
   b. Radiation Attack
   c. Other Environmental Effect

B. Time-Dependent Effects

1. Mechanical
   a. Creep
   b. Mechanical Fatigue
   c. Other Mech. Time-Dep. Effect

2. Thermal
   a. Thermal Aging
   b. Thermal Fatigue
   c. Other Thermal Time-Dep. Effect

3. Other Time-Dependent Effects
   a. Corrosion
   b. Seasonal Attack
   c. Other Time-Dep. Effect

The considerable scatter of experimental data and the lack of an exact description of the underlying physical processes for the combined mechanisms of fatigue, creep, temperature variations, and so on, make it natural, if not necessary to consider probabilistic models for a strength degradation model. Therefore, the fixed and flexible models corresponding to equation (1) are "randomized", and yield the "random normalized material strength due to a number of diverse random effects or primitive variables." Note that for the fixed model, equation (1) has the following form:

$$S/S_O = f(A_{1F}, A_1, A_{1O}, A_{2F}, A_2, A_{2O}, \ldots, A_{7F}, A_7, A_{7O}) \qquad (3)$$

where A_i, A_{iF} and A_{iO} are the ultimate, current and reference values of the i^{th} of seven effects or primitive variables as given in Table 1. In general, this expression can be written as,

$$S/S_O = f(X_i),\ i = 1, \ldots, 28, \qquad (4)$$

where the X_i are the twenty eight independent variables in equation (3). Thus, the fixed model is "randomized" by assuming all the independent variables, X_i, $i = 1, \ldots, 28$, to be random and stochastically independent. For the flexible model, equation (1) has a form analogous to equations (3) and (4), except that there are as many as seventy-two independent variables. Applying probabilistic analysis to either of these randomized equations yields the distribution of the dependent random variable, normalized material strength, S/S_O.

Although a number of methods of probabilistic analysis are available, [8] simulation was chosen for PROMISS. Simulation utilizes a theoretical sample generated by numerical techniques for each of the independent random variables. One value from each sample is substituted into the functional relationship, equation (3), and one realization of normalized strength, S/S_O, is calculated. This calculation is repeated for each value in the set of samples, yielding a distribution of different values for normalized strength.

A probability distribution function is generated from these different values of normalized strength, using a non-parametric method, maximum penalized likelihood. Maximum penalized likelihood generates the p.d.f. estimate using the method of maximum likelihood together with a penalty function to smooth it [9]. Finally, integration of the generated p.d.f. results in the cumulative distribution function, from which probabilities of normalized strength can be directly observed.

PROMISS includes computational algorithms for both the fixed and the flexible probabilistic constitutive models. As described above, PROMISS randomizes the following equation:

$$\frac{S}{S_O} = \prod_{i=1}^{n} \left[\frac{A_{iF} - A_i}{A_{iF} - A_{iO}} \right]^{a_i}, \qquad (5)$$

where

$$\left[\frac{A_{iF} - A_i}{A_{iF} - A_{iO}}\right]^{a_i}$$

is the i^{th} effect, A_i, A_{iF} and A_{iO} are random variables, a_i is the i^{th} empirical material constant and S/S_O is normalized strength. There are a maximum of eighteen possible effects or primitive variables that may be included in the model. For the flexible model option, they may be chosen by the user from those in Table 2. For the fixed model option, the primitive variables of Table 1 are chosen. Within each primitive variable term the current, ultimate and reference values and the empirical material constant may be modeled as either deterministic (empirical, calculated by PROMISC), normal, lognormal, or Wiebull random variables. Simulation is used to generate a set of realizations for normalized random strength, S/S_O, from a set of realizations for primitive variables and empirical material constants. Maximum penalized likelihood is used to generate an estimate for the p.d.f. of normalized strength, from a set of realizations of normalized strength. Integration of the p.d.f. yields the c.d.f. Plot files are produced to plot both the p.d.f. and the c.d.f. PROMISS also provides information on S/S_O statistics (mean, variance, standard deviation and coefficient of variation). A resident database, for database rather than user input of empirical material constants, is also provided.

PROMISC (see Final Technical Report, APPENDIX 2) performs a multiple linear regression on actual experimental or simulated experimental data for as many as eighteen effects or primitive variables, yielding regression coefficients that are the empirical material constants, a_i, required by PROMISS. It produces the multiple linear regression of the log transformation of equation (3), the PROMISS equation. When transformed it becomes

$$\log \frac{S}{S_O} = \sum_{i=1}^{18} a_i \log\left[\frac{A_{iF} - A_i}{A_{iF} - A_{iO}}\right], \qquad (6)$$

or

$$\log S = \log S_O + \sum_{i=1}^{18} a_i \log\left[\frac{A_{iF} - A_i}{A_{iF} - A_{iO}}\right], \qquad (7)$$

where

$$\left[\frac{A_{iF} - A_i}{A_{iF} - A_{iO}}\right]^{a_i}$$

is the i^{th} effect, A_i, A_{iF} and A_{iO} are primitive variable data and a_i is the i^{th} empirical material constant, or the i^{th} regression coefficient to be predicted by PROMISC. Also, $\log S_O$ is the log transformed reference value of strength, or the intercept regression coefficient to be predicted by PROMISC, and $\log S$ is the log transformed strength.

Experimental data for up to eighteen possible effects, as given in Table 2, may be included. The primitive variable data may be either actual experimental data or expert opinion, directly read from input, or simulated data where expert opinion is specified as the mean and standard deviation of a normal or lognormal distribution. The simulated data option for input data was used in the early stages of code development to verify correct performance. The input data, whether actual or simulated, is read in and assembled into a data matrix. From this data matrix, a corrected sums of squares and crossproducts matrix is computed. From this sums of squares and crossproducts matrix, and a least squares methodology, a multiple linear regression is performed to calculate estimates for the empirical material constant, a_i, and the reference strength, S_O. These are the regression coefficients.

PROMISC includes enhancements of the multiple linear regression analysis to screen data from "outliers" and collinearities, determine "how well" the data fit the regression, quantify the importance and relative importance of each factor in the postulated constitutive equation, eq. (1), as well as check assumptions inherent in the use of multiple linear regression. Further details are provided in the Final Technical Report, Section 6.0, NASA Grant No. NAG 3-867, Supp. 2, "Probabilistic Lifetime Strength of Aerospace Materials via Computational Simulation."

3.0 SAMPLE PROBLEMS, DATA INPUT, AND DISCUSSION OF RESULTS

Data input for PROMISS is user friendly and easy to enter. Data can be directly input by the user or the user can select data to be input by the program from its own resident database. Several examples follow (see also Section 6.0, APPENDIX B).

3.1 PROMISS Input for Flexible Model with No Statistical Distribution (Deterministic) and No use of the Resident Database

The 1st line of input (format 2E12.4, see item 1, below) determines the random number generator seed and the data sample size. The 2nd line (format I3, see item 2, below) determines either a fixed or a flexible model. The 3rd through 8th lines of input (format I3,2X,I3,2X,I3, see item 3, below) choose the 18 effects for the flexible model. On the 9th line of input (format I3, see item 4, below), the user chooses if the data is to be read from the user input or the database. The 10th line (format I3,2X,I3,2X,I3, 2X,I3, see item 4b below) determines the nature of the effect or primitive variable (deterministic or random). The 11th through 14th lines (format 10X,D12.4, see items 4b to 22, below) specify values for the effect or primitive variable. This sequence of five input lines (i.e., 9 to 14) repeats for the other effects selected (see items 5 to 21). A table listing the primitive variables, their units and symbols is given in Section 5.0, APPENDIX A. Finally, IMSL subroutine parameters are entered as indicated in items 23 and 24.

1. Line 1 selects the Random Number Generator Seed (ISEED) and Sample Size (NTOT).

EXAMPLE:

```
12345678901234567890*
  1     40
```

2. Line 2 selects either Fixed or Flexible Model (MODEL = 0 is flexible, MODEL = 1 is fixed).

EXAMPLE:

```
12345678901234567890
  0
```

3. Lines 3 to 8 select the 18 effects for the Flexible Model where:

EFFMS = 1 is Quasi-static Stress Effect
EFFMS = 0 is No Quasi-static Stress Effect
EFFMI = 1 is Impact Effect
EFFMI = 0 is No Impact Effect

* NOTE: the ruler is to aid the user in formatting and is not a part of the input.

EFFMO = 1 is Other Mechanical Effect
EFFMO = 0 is No Other Mechanical Effect

EFFTT = 1 is Temperature Effect
EFFTT = 0 is No Temperature Effect
EFFTS = 1 is Thermal Shock Effect
EFFTS = 0 is No Thermal Shock Effect
EFFTO = 1 is Other Thermal Effect
EFFTO = 0 is No Other Thermal Effect

EFFOC = 1 is Chemical Effect
EFFOC = 0 is No Chemical Effect
EFFOR = 1 is Radiation Effect
EFFOR = 0 is No Radiation Effect
EFFOO = 1 is Other Effect
EFFOO = 0 is No Other Effect

TEFFMC = 1 is Creep Effect
TEFFMC = 0 is No Creep Effect
TEFFMF = 1 is Mechanical Fatigue Effect
TEFFMF = 0 is No Mechanical Fatigue Effect
TEFFMO = 1 is Other Time-Dependent Mechanical Effect
TEFFMO = 0 is No Other Time-Dependent Mechanical Effect

TEFFTA = 1 is Thermal Aging Effect
TEFFTA = 0 is No Thermal Aging Effect
TEFFTF = 1 is Thermal Fatigue Effect
TEFFTF = 0 is No Thermal Fatigue Effect
TEFFTO = 1 is Other Time-Dependent Thermal Effect
TEFFTO = 0 is No Other Time-Dependent Thermal Effect

TEFFOC = 1 is Corrosion Effect
TEFFOC = 0 is No Corrosion Effect
TEFFOS = 1 is Seasonal Attack Effect
TEFFOS = 0 is No Seasonal Attack Effect
TEFFOO = 1 is Other Time-Dependent Effect
TEFFOO = 0 is No Other Time-Dependent Effect.

The effects are read from the input file as follows:

EFFMS, EFFMI, EFFMO
EFFTT, EFFTS, EFFTO
EFFOC, EFFOR, EFFOO
TEFFMC, TEFFMF, TEFFMO
TEFFTA, TEFFTF, TEFFTO
TEFFOC, TEFFOS, TEFFOO.

EXAMPLE:

```
12345678901234567890
    1       1       1
    1       1       1
    1       1       1
    1       1       1
    1       1       1
    1       1       1
```

4a. Line 9 selects either User Input or Resident Database for the Exponent for Quasi-static Stress Effect (DATA = 0 is user input , DATA = 1 is database).

EXAMPLE:

```
12345678901234567890
    0
```

4b. Line 10 specifies the *nature* of the four variables within the effect by using flags. The flag names are AFDS, ADS, AODS, and SADS. Setting a flag to 0 indicates a deterministic or nominal value will be input for the variable. Setting a flag to 1 indicates a normal distribution. Setting a flag to 2 indicates a lognormal distribution. Setting a flag to 3 indicates a special distribution not yet developed. Setting a flag to 4 indicates a Weibull distribution. Lines 11 to 14 give the values of the four variables. The Quasi-static Stress Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
    0       0       0       0
                        5.0D+00
                      3.855D+00
                        4.0D+00
                        1.0D+00
```

5. Lines 15 to 20 input data for the Impact Effect in the same manner as was described for the Quasi-static Stress Effect (see items 4a and 4b, above). The Impact Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
    0
    0       0       0       0
                        5.0D+00
                      3.855D+00
                        4.0D+00
                        1.0D+00
```

6. Lines 21 to 26 input data for the Other Mechanical Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Mechanical Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
    0
    0         0         0         0
                                  5.0D+00
                              3.855D+00
                                  4.0D+00
                                  1.0D+00
```

7. Lines 27 to 32 input data for the Temperature Effect in the same manner as was described for the Quasi-static Stress Effect. The Temperature Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
    0
    0         0         0         0
                                  5.0D+00
                              3.855D+00
                                  4.0D+00
                                  1.0D+00
```

8. Lines 33 to 38 input data for the Thermal Shock Effect in the same manner as was described for the Quasi-static Stress Effect. The Thermal Shock Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
    0
    0         0         0         0
                                  5.0D+00
                              3.855D+00
                                  4.0D+00
                                  1.0D+00
```

9. Lines 39 to 44 input data for the Other Thermal Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Thermal Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901
    0
    0       0       0       0
                         5.0D+00
                      3.855D+00
                         4.0D+00
                         1.0D+00
```

10. Lines 45 to 50 input data for the Chemical Effect in the same manner as was described for the Quasi-static Stress Effect. The Chemical Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901
    0
    0       0       0       0
                         5.0D+00
                      3.855D+00
                         4.0D+00
                         1.0D+00
```

11. Lines 51 to 56 input data for the Radiation Effect in the same manner as was described for the Quasi-static Stress Effect. The Radiation Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901
    0
    0       0       0       0
                         5.0D+00
                      3.855D+00
                         4.0D+00
                         1.0D+00
```

12. Lines 57 to 62 input data for the Other Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
  0
  0      0      0      0
                   5.0D+00
                 3.855D+00
                   4.0D+00
                   1.0D+00
```

13. Lines 63 to 68 input data for the Creep Effect in the same manner as was described for the Quasi-static Stress Effect. The Creep Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
  0
  0      0      0      0
                   5.0D+00
                 3.855D+00
                   4.0D+00
                   1.0D+00
```

14. Lines 69 to 74 input data for the Mechanical Fatigue Effect in the same manner as was described for the Quasi-static Stress Effect. The Mechanical Fatigue Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
  0
  0      0      0      0
                   5.0D+00
                 3.855D+00
                   4.0D+00
                   1.0D+00
```

15. Lines 75 to 80 input data for the Time-dependent Mechanical Effect in the same manner as was described for the Quasi-static Stress Effect. The Time-dependent Mechanical Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
   0
   0       0       0       0
                             5.0D+00
                           3.855D+00
                             4.0D+00
                             1.0D+00
```

16. Lines 81 to 86 input data for the Thermal Aging Effect in the same manner as was described for the Quasi-static Stress Effect. The Thermal Aging Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
   0
   0       0       0       0
                             5.0D+00
                           3.855D+00
                             4.0D+00
                             1.0D+00
```

17. Lines 87 to 92 input data for the Thermal Fatigue Effect in the same manner as was described for the Quasi-static Stress Effect. The Thermal Fatigue Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
   0
   0       0       0       0
                             5.0D+00
                           3.855D+00
                             4.0D+00
                             1.0D+00
```

18. Lines 93 to 98 input data for the Other Time-dependent Thermal Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Time-dependent Thermal Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
   0
   0       0       0       0
                        5.0D+00
                     3.855D+00
                        4.0D+00
                        1.0D+00
```

19. Lines 99 to 104 input data for the Corrosion Effect in the same manner as was described for the Quasi-static Stress Effect. The Corrosion Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
   0
   0       0       0       0
                        5.0D+00
                     3.855D+00
                        4.0D+00
                        1.0D+00
```

20. Lines 105 to 110 input data for the Seasonal Attack Effect in the same manner as was described for the Quasi-static Stress Effect. The Seasonal Attack Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
   0
   0       0       0       0
                        5.0D+00
                     3.855D+00
                        4.0D+00
                        1.0D+00
```

21. Lines 111 to 116 input data for the Other Time-dependent Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Time-dependent Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
123456789012345678901234567890
  0
  0    0    0    0
                5.0D+00
              3.855D+00
                4.0D+00
                1.0D+00
```

22. The DESPL [1] parameters are NODE, INIT, ALPHA, EPS, and MAXIT and are entered in that order as follows (line 117):

EXAMPLE:

```
1234567890123456789012345678901234567890
 21     0         1.0E+01     1.0E-05   30
```

23. The DESPL[1] parameter, IOPT, is entered as follows (line 118):

EXAMPLE:

```
1234567890
  2
```

3.1.1 Discussion of Results

Execution of PROMISS (source code entitled PROMISS89.FOR) produces an output file that gives numerical results (see Section 6.0, APPENDIX B). Execution also produces plotfiles (see Section 6.0, APPENDIX B). These files are used to plot the X and Y axes of the probability density function (p.d.f.) and the cumulative distribution function (c.d.f.) generated by PROMISS. The plots are drawn from the plotfiles by the SAS/GRAPH graphing program (see Section 8.0, APPENDIX D).

3.2 PROMISS Input for Fixed Model with Various Statistical Distributions and use of the Resident Database

The 1st line of input (format 2E12.4, see item 1, below) determines the random number generator seed and the data sample size. The 2nd line (format I3, see item 2, below) determines either a fixed or a flexible model. On the 3rd line of input the user chooses if the data is to be read from the user input or the resident database (format I3, see item 3, below). The 4th line (format I3,2X,I3,2X,I3, 2X,I3, see item 4, below) determines the nature of the effect or primitive variable (deterministic or random). The 5th through 8th lines specify values for the effect or primitive variable. This sequence of five input lines repeats for the other effects (see items 5 through 10, below). If the word DATABASE is entered in the place of the fourth integer (format I3,2X,I3,2X,I3,2X,A8), the program will read a deterministic value for the exponent from its own Resident Database. A table listing the primitive variables, their units and symbols is given in APPENDIX A. Finally, IMSL subroutine parameters are entered as indicated in items 12 and 13.

1. Line 1 selects Random Number Generator Seed (ISEED) and Sample Size (NTOT).

EXAMPLE:

```
12345678901234567890*
    1        40
```

2. Line 2 determines fixed or flexible model (MODEL = 0 is flexible, MODEL = 1 is fixed).

EXAMPLE:

```
12345678901234567890
    1
```

3. Line 3 determines User Input or Database for the Exponent for Quasi-Static Stress Effect (DATA = 0 is user input, DATA = 1 is database).

EXAMPLE:

```
12345678901234567890
    1
```

* NOTE: the ruler is to aid the user in formatting and is not a part of the input.

4. Line 4 specifies the *nature* of the four variables within the effect by using flags. The flag names are AFDS, ADS, AODS, and SADS. Setting a flag to 0 indicates a deterministic or nominal value will be input for the variable. Setting a flag to 1 indicates a normal distribution. Setting a flag to 2 indicates a lognormal distribution. Setting a flag to 3 indicates a special distribution not yet developed. Setting a flag to 4 indicates a Weibull distribution. Lines 5 to 7 give the values of the four variables. The Quasi-static Stress Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
   4     4     2   DATABASE
                  130.0D+00    6.500D+00
                   90.0D+00    4.500D+00
                   -2.9D+00   -0.145D+00
```

5. Line 8 determines user input or database for the exponent for Impact Effect. Lines 9 to 13 input data for the Impact effect in the same manner as was described for the Quasi-static Stress Effect (see items 3 and 4, above). The Impact Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
   0
   4     1     2     4
                    1.0D+00    0.003D+00
                   0.10D+00    0.003D+00
                  0.001D+00 0.00003D+00
                   -0.5D+00   -0.015D+00
```

6. Line 14 determines user input or database for the exponent for Temperature Effect. Lines 15 to 19 input data for the Temperature Effect in the same manner as was described for the Quasi-static Stress Effect. The Temperature Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1244567890123456789012345678901234567890
   0
   2     2     1     2
                 2732.0D+00   82.000D+00
                 1562.0D+00    46.70D+00
                   68.0D+00    2.040D+00
                   0.50D+00    0.015D+00
```

7. Line 20 determines user input or database for the exponent for Chemical Reaction Effect. Lines 21 to 25 input data for the Chemical Reaction Effect in the same manner as was described for the Quasi-static Stress Effect. The Chemical Reaction Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
 0
 1    1    1    1
                 1.0D+00    0.003D+00
                0.02D+00   0.0006D+00
               0.001D+00  0.00003D+00
                0.50D+00    0.015D+00
```

8. Line 26 determines user input or database for the exponent for Creep Effect. Lines 27 to 30 input data for the Creep Effect in the same manner as was described for the Quasi-static Stress Effect. The Creep Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A. Note that when the DATABASE option is used, the input is only 5 lines long per effect, rather than 6 lines (compare to item 7, above).

EXAMPLE:

```
1234567890123456789012345678901234567890
 1
 2    4    2  DATABASE
             10000.0D+00  500.00D+00
               105.0D+00     3.15D+00
               0.083D+00   0.0025D+00
```

9. Line 31 determines user input or database for the exponent for Mechanical Fatigue Effect. Lines 32 to 36 input data for the Mechanical Fatigue Effect in the same manner as was described for the Quasi-static Stress Effect. The Mechanical Fatigue Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
 0
 2    2    2    1
                 7.0D+00    0.700D+00
                 3.5D+00    0.350D+00
                 1.0D+00    0.100D+00
                0.50D+00    0.015D+00
```

10. Line 37 determines user input or database for the exponent for Thermal Fatigue Effect. Lines 38 to 41 input data for the Thermal Fatigue Effect in the same manner as was described for the Quasi-static Stress Effect. The Thermal Fatigue Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  2      2      2   DATABASE
                    3.0D+00     0.300D+00
                    2.3D+00     0.230D+00
                    1.0D+00     0.100D+00
```

11. The DESPL[1] parameters are NODE, INIT, ALPHA, EPS, and MAXIT and are entered in that order as follows (line 42):

EXAMPLE:

```
12345678901234567890123456789012345012345678901234567890
 21     0          1.0E+01      1.0E-05    30
```

12. The DESPL[1] parameter, IOPT, is entered as follows (line 43):

EXAMPLE:

```
1234567890
  2
```

DATABASE INPUT FILE**

```
12345678901234567890
     0.5D+00
     0.5D+00
     0.5D+00
```

3.2.1 Discussion of Results

Execution of PROMISS (source code entitled PROMISS89.FOR) produces an output file that gives numerical results (see Section 6.0, APPENDIX B). Execution also produces plotfiles (see Section 6.0, APPENDIX B). These files are used to plot the X and Y axes of the probability density function (p.d.f.) and the cumulative distribution function (c.d.f.) generated by PROMISS. The plots are drawn from the plotfiles by the SAS/GRAPH graphing program (see Section 8.0, APPENDIX D). These plots for the sample problem are shown in Figures 2 and 3.

** PROMISS, Vers. 2.0, currently has a value of 0.5 for all emperical material constants. This is current expert opinion. These values will be updated as data become available (see Section 6.0).

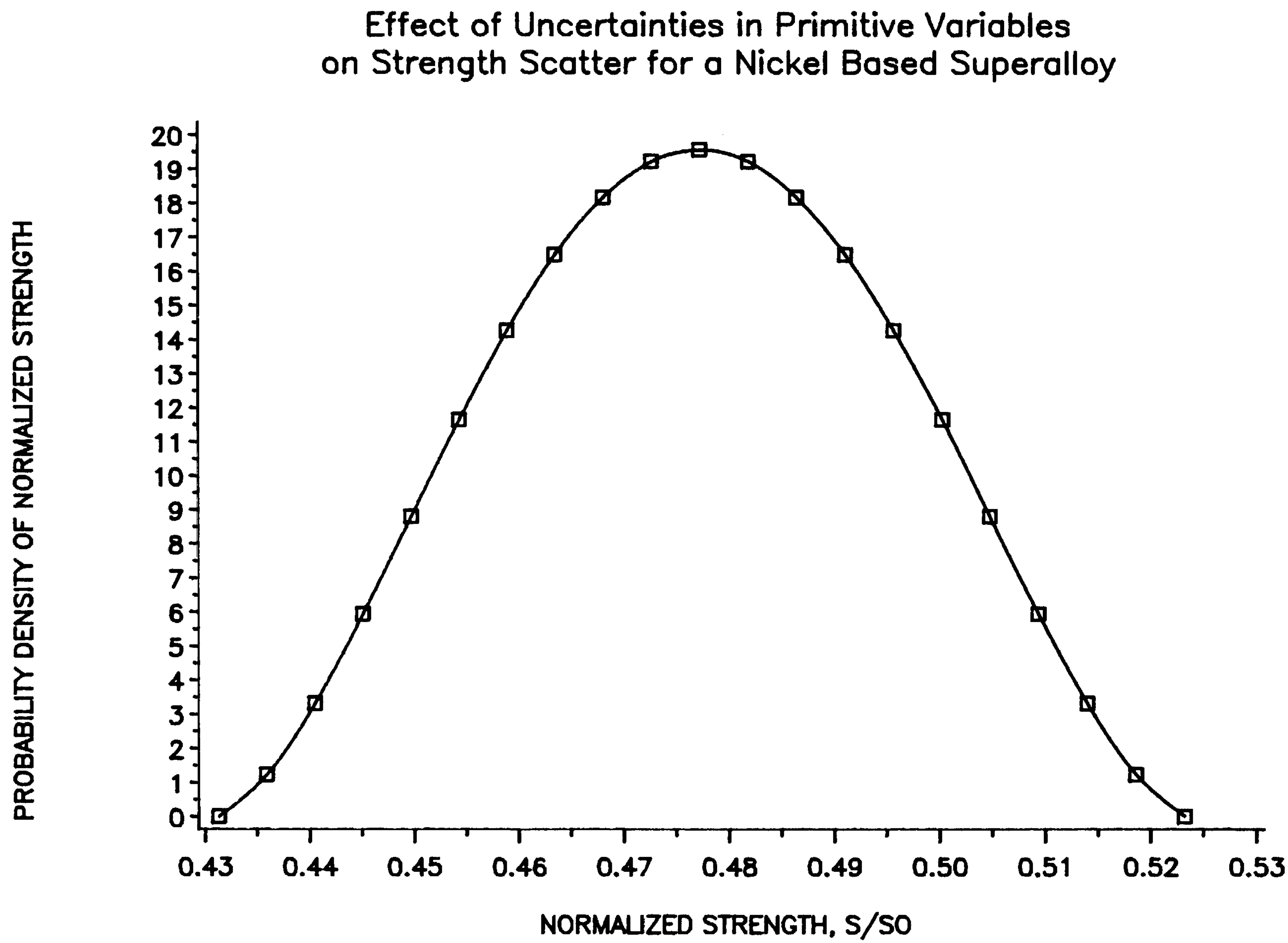

Figure 2 Section 3.2 Example Plot Created Using SAS/GRAPH

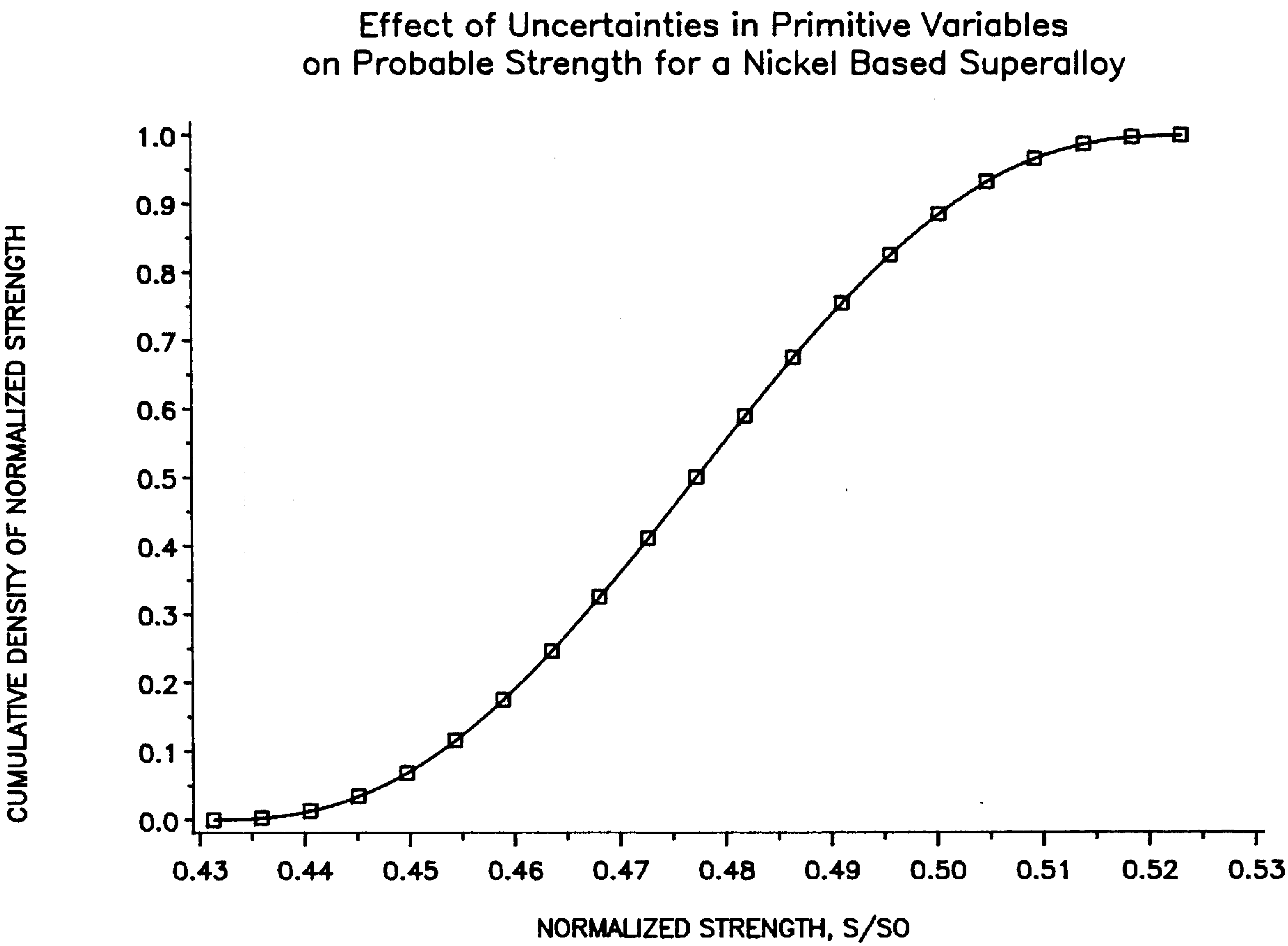

Figure 3 Section 3.2 Example Plot Created Using SAS/GRAPH

3.3 PROMISS Input for Flexible Model with Various Statistical Distributions and Use of the Resident Database

The 1st line of input (format 2E12.4, see item 1, below) determines the random number generator seed and the data sample size. The 2nd line (format I3, see item 2, below) determines either a fixed or a flexible model. The 3rd through 8th lines (format I3,2X,I3,2X,I3, see item 3, below) choose the 18 effects for the flexible model. The 9th line (format I3, see item 4, below) the user chooses if the data is to be read from the user input or the database. The 10th line (format I3,2X,I3,2X,I3, 2X,I3, see item 4, below) determines the nature of the effect or primitive variable. If the word DATABASE is entered in the place of the fourth integer (format I3,2X,I3,2X,I3,2X,A8), the program will read a deterministic value from its own Resident Database. The 11th through 14th lines (format 10X,2D12.4, see item 4, below) specify values for the effect or primitive variable. This sequence of five input lines repeats for the other effects selected (see items 4 through 21, below). A table listing the primitive variables, their units and symbols is given in Section 5.0, APPENDIX A. Finally, IMSL subroutine parameters are entered as indicated in items 23 and 24.

1. Line 1 selects the Random Number Generator Seed (ISEED) and Sample Size (NTOT).

EXAMPLE:

```
12345678901234567890*
   1       40
```

2. Line 2 determines fixed or flexible model (MODEL = 0 is flexible, MODEL = 1 is fixed).

EXAMPLE:

```
12345678901234567890
   0
```

3. Lines 3 to 8 select the 18 Effects for the Flexible Model where:

EFFMS = 1 is Quasi-static Stress Effect
EFFMS = 0 is No Quasi-static Stress Effect
EFFMI = 1 is Impact Effect
EFFMI = 0 is No Impact Effect
EFFMO = 1 is Other Mechanical Effect
EFFMO = 0 is No Other Mechanical Effect

EFFTT = 1 is Temperature Effect
EFFTT = 0 is No Temperature Effect
EFFTS = 1 is Thermal Shock Effect
EFFTS = 0 is No Thermal Shock Effect
EFFTO = 1 is Other Thermal Effect
EFFTO = 0 is No Other Thermal Effect

* NOTE: the ruler is to aid the user in formatting and is not a part of the input.

EFFOC = 1 is Chemical Effect
EFFOC = 0 is No Chemical Effect
EFFOR = 1 is Radiation Effect
EFFOR = 0 is No Radiation Effect
EFFOO = 1 is Other Effect
EFFOO = 0 is No Other Effect

TEFFMC = 1 is Creep Effect
TEFFMC = 0 is No Creep Effect
TEFFMF = 1 is Mechanical Fatigue Effect
TEFFMF = 0 is No Mechanical Fatigue Effect
TEFFMO = 1 is Other Time-Dependent Mechanical Effect
TEFFMO = 0 is No Other Time-Dependent Mechanical Effect

TEFFTA = 1 is Thermal Aging Effect
TEFFTA = 0 is No Thermal Aging Effect
TEFFTF = 1 is Thermal Fatigue Effect
TEFFTF = 0 is No Thermal Fatigue Effect
TEFFTO = 1 is Other Time-Dependent Thermal Effect
TEFFTO = 0 is No Other Time-Dependent Thermal Effect

TEFFOC = 1 is Corrosion Effect
TEFFOC = 0 is No Corrosion Effect
TEFFOS = 1 is Seasonal Attack Effect
TEFFOS = 0 is No Seasonal Attack Effect
TEFFOO = 1 is Other Time-Dependent Effect
TEFFOO = 0 is No Other Time-Dependent Effect.

The effects are read from the input file as follows:

EFFMS, EFFMI, EFFMO
EFFTT, EFFTS, EFFTO
EFFOC, EFFOR, EFFOO
TEFFMC, TEFFMF, TEFFMO
TEFFTA, TEFFTF, TEFFTO
TEFFOC, TEFFOS, TEFFOO.

EXAMPLE:

```
12345678901234567890
    1    1    1
    1    1    1
    1    1    1
    1    1    1
    1    1    1
    1    1    1
```

4. Line 9 selects either User Input or Resident Database for the Exponent for Quasi-static Stress Effect (DATA = 0 is user input , DATA = 1 is database). Line 10 specifies the *nature* of the four variables within the effect by using flags. The flag names are AFDS, ADS, AODS, and SADS. Setting a flag to 0 indicates a deterministic or nominal value will be input for the variable. Setting a flag to 1 indicates a normal distribution. Setting a flag to 2 indicates a lognormal distribution. Setting a flag to 3 indicates a special distribution not yet developed. Setting a flag to 4 indicates a Weibull distribution. Lines 11 to 14 give the values of the four variables. The Quasi-static Stress Effect variables names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
    0
    2    4    4    1
                        5.0D+00    0.002D+00
                      3.855D+00    0.005D+00
                        4.0D+00    0.007D+00
                       0.51D+00    0.001D+00
```

5. Line 15 determines user input or database for the exponent for Impact Effect. Lines 16 to 19 input data for the Impact effect in the same manner as was described for the Quasi-static Stress Effect (see items 3 and 4, above). The Impact Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A. Note that when the DATABASE option is used, the input is 5 lines long per effect, rather than 6 lines (compare to item 4 above).

EXAMPLE:

```
1234567890123456789012345678901234567890
    1
    4    1    2    DATABASE
                        5.0D+00    0.002D+00
                      3.855D+00    0.005D+00
                        4.0D+00    0.007D+00
```

6. Line 20 determines user input or database for the exponent for Mechanical Effect. Lines 21 to 24 input data for the Mechanical Effect in the same manner as was described for the Quasi-static Stress Effect. The Mechanical Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
    1
    4    4    2    DATABASE
                        5.0D+00    0.002D+00
                      3.855D+00    0.005D+00
                        4.0D+00    0.007D+00
```

7. Line 25 determines user input or database for the exponent for Temperature Effect. Lines 26 to 29 input data for the Temperature Effect in the same manner as was described for the Quasi-static Stress Effect. The Temperature Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  1    1    4  DATABASE
                5.0D+00   0.002D+00
              3.855D+00   0.005D+00
                4.0D+00   0.007D+00
```

8. Line 30 determines user input or database for the exponent for Thermal Shock Effect. Lines 31 to 35 input data for the Thermal Shock Effect Effect in the same manner as was described for the Quasi-static Stress Effect. The Thermal Shock Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  0
  4    2    1    2
                5.0D+00   0.002D+00
              3.855D+00   0.005D+00
                4.0D+00   0.007D+00
               0.51D+00   0.001D+00
```

9. Line 36 determines user input or database for the exponent for Other Thermal Effect. Lines 37 to 40 input data for the Other Thermal Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Thermal Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  4    2    1  DATABASE
                5.0D+00   0.002D+00
              3.855D+00   0.005D+00
                4.0D+00   0.007D+00
```

10. Line 41 determines user input or database for the exponent for Chemical Reaction Effect. Lines 42 to 45 input data for the Chemical Reaction Effect in the same manner as was described for the Quasi-static Stress Effect. The Chemical Reaction Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
 1
 4    1    4  DATABASE
               5.0D+00   0.002D+00
             3.855D+00   0.005D+00
               4.0D+00   0.007D+00
```

11. Line 46 determines user input or database for the exponent for Radiation Effect. Lines 47 to 50 input data for the Radiation Effect in the same manner as was described for the Quasi-static Stress Effect. The Radiation Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
 1
 4    1    4  DATABASE
               5.0D+00   0.002D+00
             3.855D+00   0.005D+00
               4.0D+00   0.007D+00
```

12. Line 51 determines user input or database for the exponent for Other Effect. Lines 52 to 55 input data for the Other Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
 1
 1    4    2  DATABASE
               5.0D+00   0.002D+00
             3.855D+00   0.005D+00
               4.0D+00   0.007D+00
```

13. Line 56 determines user input or database for the exponent for Creep Effect. Lines 57 to 60 input data for the Creep Effect in the same manner as was described for the Quasi-static Stress Effect. The Creep Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A. The last line for this Effect is eliminated when the database is used.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  1    2    2   DATABASE
                 5.0D+00    0.002D+00
               3.855D+00    0.005D+00
                 4.0D+00    0.007D+00
```

14. Line 61 determines user input or database for the exponent for Mechanical Fatigue Effect. Lines 62 to 66 input data for the Mechanical Fatigue Effect in the same manner as was described for the Quasi-static Stress Effect. The Mechanical Fatigue Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  0
  1    1    2    1
                 5.0D+00    0.002D+00
               3.885D+00    0.005D+00
                 4.0D+00    0.007D+00
                0.51D+00    0.001D+00
```

15. Line 67 determines user input or database for the exponent for Other Time-dependent Mechanical Effect. Lines 68 to 71 input data for the Other Time-dependent Mechanical Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Time-dependent Mechanical Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  2    2    2   DATABASE
                 5.0D+00    0.002D+00
               3.855D+00    0.005D+00
                 4.0D+00    0.007D+00
```

16. Line 72 determines user input or database for the exponent for Thermal Aging Effect. Lines 73 to 76 input data for the Thermal Aging Effect in the same manner as was described for the Quasi-static Stress Effect. The Thermal Aging Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  1     1     1  DATABASE
                 5.0D+00    0.002D+00
               3.855D+00    0.005D+00
                 4.0D+00    0.007D+00
```

17. Line 77 determines user input or database for the exponent for Thermal Fatigue Effect. Lines 78 to 81 input data for the Thermal Fatigue Effect in the same manner as was described for the Quasi-static Stress Effect. The Thermal Fatigue Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  4     4     4  DATABASE
                 5.0D+00    0.002D+00
               3.855D+00    0.005D+00
                 4.0D+00    0.007D+00
```

18. Line 82 determines user input or database for the exponent for Other Time-dependent Thermal Effect. Lines 83 to 86 input data for the Other Time-dependent Thermal Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Time-dependent Thermal Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A. The last line for this Effect is eliminated when the database is used.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  4     2     2  DATABASE
                 5.0D+00    0.002D+00
               3.855D+00    0.005D+00
                 4.0D+00    0.007D+00
```

19. Line 87 determines user input or database for the exponent for Corrosion Effect. Lines 88 to 91 input data for the Corrosion Effect in the same manner as was described for the Quasi-static Stress Effect. The Corrosion Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  1     1     4  DATABASE
                   5.0D+00    0.002D+00
                 3.855D+00    0.005D+00
                   4.0D+00    0.007D+00
```

20. Line 92 determines user input or database for the exponent for Seasonal Attack Effect. Lines 93 to 96 input data for the Seasonal Attack Effect in the same manner as was described for the Quasi-static Stress Effect. The Seasonal Attack Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  2     4     4  DATABASE
                   5.0D+00    0.002D+00
                 3.855D+00    0.005D+00
                   4.0D+00    0.007D+00
```

21. Line 97 determines user input or database for the exponent for Other Time-dependent Effect. Lines 98 to 101 input data for the Other Time-dependent Effect in the same manner as was described for the Quasi-static Stress Effect. The Other Time-dependent Effect variable names have FORTRAN names given in Section 5.0, APPENDIX A.

EXAMPLE:

```
1234567890123456789012345678901234567890
  1
  4     2     4  DATABASE
                   5.0D+00    0.002D+00
                 3.855D+00    0.005D+00
                   4.0D+00    0.007D+00
```

22. The DESPL[1] parameters are NODE, INIT, ALPHA, EPS, and MAXIT and are entered in that order as follows (line 102):

EXAMPLE:

```
1234567890123456789012345678901234567890
 21     0         1.0E+01     1.00E-05   30
```

23. The DESPL[1] parameter, IOPT, is entered as follows (line 103):

EXAMPLE:

```
1234567890
  2
```

DATABASE INPUT FILE

```
12345678901234567890
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
    0.5D+00
```

3.3.1 Discussion of Results

Execution of PROMISS (source code entitled PROMISS89.FOR) produces an output file that gives numerical results (see Section 6.0, APPENDIX B). Execution also produces plotfiles (see Section 6.0, APPENDIX B). These files are used to plot the X and Y axes of the probability density function (p.d.f.) and the cumulative distribution function (c.d.f.) generated by PROMISS. The plots are drawn from the plotfiles by the SAS/GRAPH graphing program (see Section 8.0, APPENDIX D).

3.4 General PROMISS Program Notes

1. Normalized strength, S/SO, is assured to be non-negative; any negative value calculated from input is set arbitrarily to zero.

2. Any effect, $\left[\frac{A_F - A}{A_F - A_O}\right]_i^{a_i}$, not included in the model is set to one (1).

3. The IMSL, Vers. 10 subroutine DESPL requires that IMSL subroutine D3SPL be appended to PROMISS for proper operation.

4. Fatigue cycles should be input as log cycles rather than cycles. This assures fatigue affects strength calculations by yielding a fraction significantly below zero. If input for fatigue was in cycles, the value of the effect approaches one, thereby not affecting strength calculations.

4.0 REFERENCES

1. IMSL, "STAT/LIBRARY, FORTRAN Subroutines for Statistical Analysis", Houston, Texas.

2. SAS Institute, Inc., SAS/GRAPH User's Guide, Version 5 Edition. Cary, NC: SAS Institute, Inc., 1985, p. 596.

3. Chamis, C.C., "Simplified Composite Micromechanics Equations for Strength, Fracture Toughness, Impact Resistance and Environmental Effects," NASA TM 83696, Jan., 1984.

4. Hopkins, D.A., "Nonlinear Analysis for High-Temperature Multilayered Fiber Composite Structures," NASA TM 83754, Aug., 1984.

5. Chamis, C.C. and Hopkins, D., "Thermoviscoplastic Nonlinear Constitutive Relationships for Structural Analysis of High Temperature Metal Matrix Composites," NASA TM 87291, Nov., 1985.

6. Hopkins, D. and Chamis, C.C., "A Unique Set of Micromechanics Equations for High Temperature Metal Matrix Composites," NASA TM 87154.

7. Ross, S.M., Introduction to Probability and Statistics for Engineers and Scientists, Wiley, N.Y., 1987, p.278.

8. Siddall, J.N., "A Comparison of Several Methods of Probabilistic Modeling," Proceedings of the Computers in Engineering Conference, ASME San Diego, CA, Vol. 4, 1982, pp. 231-238.

9. Scott, D.W., "Nonparametric Probability Density Estimation by Optimization Theoretic Techniques," NASA CR-147763, April, 1976.

5.0 APPENDIX A

PRIMITIVE VARIABLES, SYMBOLS, AND UNITS

Table A1.2 Primitive variables, symbols, and units for PROMISS

Primitive Variables (Effect)	Theory Symbol	FORTRAN Name	Units SI	Units U.S.
QUASI-STATIC STRESS EFFECT				
Ultimate value	S_{SF}	AF	MPa	ksi
Current value	σ_S	A	MPa	ksi
Reference value	σ_{SO}	AO	MPa	ksi
Material constant	p	SA	Dimensionless	
IMPACT EFFECT				
Ultimate value	S_{DF}	AF2	Dimensionless	
Current value	σ_D	A2	Dimensionless	
Reference value	σ_{DO}	AO2	Dimensionless	
Material constant	s	SA2	Dimensionless	
OTHER MECHANICAL EFFECTS				
Ultimate value	A_{3F}	AF3	Dimensionless	
Current value	A_3	A3	Dimensionless	
Reference value	A_{3O}	AO3	Dimensionless	
Material constant	a_3	SA3	Dimensionless	
TEMPERATURE				
Ultimate value	T_F	AF4	^oC	^oF
Current value	T	A4	^oC	^oF
Reference value	T_O	AO4	^oC	^oF
Material constant	q	SA4	Dimensionless	
THERMAL SHOCK				
Ultimate value	A_{5F}	AF5	Dimensionless	
Current value	A_5	A5	Dimensionless	
Reference value	A_{5O}	AO5	Dimensionless	
Material constant	a_5	SA5	Dimensionless	

Primitive Variables (Effect)	Theory Symbol	FORTRAN Name	Units SI	U.S.
OTHER THERMAL EFFECTS				
Ultimate value	A_{6F}	AF6	Dimensionless	
Current value	A_6	A6	Dimensionless	
Reference value	A_{6O}	AO6	Dimensionless	
Material constant	a_6	SA6	Dimensionless	
CHEMICAL EFFECTS				
Ultimate value	R_F	AF7	Dimensionless	
Current value	R	A7	Dimensionless	
Reference value	R_O	AO7	Dimensionless	
Material constant	r	SA7	Dimensionless	
RADIATION EFFECTS				
Ultimate value	A_{8F}	AF8	Dimensionless	
Current value	A_8	A8	Dimensionless	
Reference value	A_{8O}	AO8	Dimensionless	
Material constant	a_8	SA8	Dimensionless	
OTHER EFFECTS				
Ultimate value	A_{9F}	AF9	Dimensionless	
Current value	A_9	A9	Dimensionless	
Reference value	A_{9O}	AO9	Dimensionless	
Material constant	a_9	SA9	Dimensionless	
CREEP				
Ultimate value	t_{CF}	AF10	Hours	Hours
Current value	t_C	A10	Hours	Hours
Reference value	t_{CO}	AO10	Hours	Hours
Material constant	v	SA10	Dimensionless	
MECHANICAL FATIGUE				
Ultimate value	N_{MF}	AF11	log of cycles	
Current value	N_M	A11	log of cycles	
Reference value	N_{MO}	AO11	log of cycles	
Material constant	t	SA11	Dimensionless	

Primitive Variables (Effect)	Theory Symbol	FORTRAN Name	Units SI	Units U.S.
<u>OTHER TIME DEPENDENT MECHANICAL EFFECT</u>				
Ultimate value	A_{12F}	AF12	Dimensionless	
Current value	A_{12}	A12	Dimensionless	
Reference value	A_{12O}	AO12	Dimensionless	
Material constant	a_{12}	SA12	Dimensionless	
<u>THERMAL AGING</u>				
Ultimate value	A_{13F}	AF13	Dimensionless	
Current value	A_{13}	A13	Dimensionless	
Reference value	A_{13O}	AO13	Dimensionless	
Material constant	a_{13}	SA13	Dimensionless	
<u>THERMAL FATIGUE</u>				
Ultimate value	N_{TF}	AF14	log of cycles	
Current value	N_T	A14	log of cycles	
Reference value	N_{TO}	AO14	log of cycles	
Material constant	u	SA14	Dimensionless	
<u>OTHER TIME-DEPENDENT THERMAL EFFECTS</u>				
Ultimate value	A_{15F}	AF15	Dimensionless	
Current value	A_{15}	A15	Dimensionless	
Reference value	A_{15O}	AO15	Dimensionless	
Material constant	a_{15}	SA15	Dimensionless	
<u>CORROSION</u>				
Ultimate value	A_{16F}	AF16	Dimensionless	
Current value	A_{16}	A16	Dimensionless	
Reference value	A_{16O}	AO16	Dimensionless	
Material constant	a_{16}	SA16	Dimensionless	

Primitive Variables (Effect)	Theory Symbol	FORTRAN Name	Units SI	U.S.
<u>SEASONAL ATTACK</u>				
Ultimate value	A_{17F}	AF17	Dimensionless	
Current value	A_{17}	A17	Dimensionless	
Reference value	A_{17O}	AO17	Dimensionless	
Material constant	a_{17}	SA17	Dimensionless	
<u>OTHER TIME-DEPENDENT EFFECT</u>				
Ultimate value	A_{18F}	AF18	Dimensionless	
Current value	A_{18}	A18	Dimensionless	
Reference value	A_{18O}	AO18	Dimensionless	
Material constant	a_{18}	SA18	Dimensionless	

6.0 APPENDIX B

PROMISS SAMPLE PROBLEM: INPUT AND OUTPUT FILES

Sample problems are discussed in sections 3.1, 3.2, and 3.3. One sample problem corresponds to each section. The input and output file names for each sample problem are listed below. The enclosed disk also includes the same files.

3.1 PROMISS INPUT FOR FLEXIBLE MODEL WITH NO STATISTICAL DISTRIBUTION (DETERMINISTIC) AND NO USE OF THE RESIDENT DATABASE

Input File(s):	31PR.INP
Output File:	31PR.OUT
Plot Files:	31PR1.PLT
	31PR2.PLT

3.2 PROMISS INPUT FOR FIXED MODEL WITH VARIOUS STATISTICAL DISTRIBUTIONS AND USE OF THE RESIDENT DATABASE

Input File(s):	32PR1.INP
	32PR2.INP
Output File:	32PR.OUT
Plot Files:	32PR1.PLT
	32PR2.PLT

3.3 PROMISS INPUT FOR FLEXIBLE MODEL WITH VARIOUS STATISTICAL DISTRIBUTIONS AND USE OF THE RESIDENT DATABASE

Input File(s):	33PR1.INP
	33PR2.INP
Output File:	33PR.OUT
Plot Files:	33PR1.PLT
	33PR2.PLT

7.0 APPENDIX C

IMSL SUBROUTINE CALLS FROM PROMISS

PROMISS

1. RNSET - Initializes a random seed for use in the IMSL random number generators.

2. RNNOR - Generates pseudorandom numbers from a standard normal distribution using an inverse CDF method.

3. RNLNL - Generates pseudorandom numbers from a lognormal distribution.

4. RNWIB - Generates pseudorandom numbers from a Weibull distribution.

5. DESPL - Performs nonparametric probability density function estimation by the penalized likelihood method.

6. GCDF - Evaluates a general continuous cumulative distribution function given the ordinates of the density.

7. Other IMSL Subroutines - SSCAL and SADD

8.0 APPENDIX D

SAMPLE SAS/GRAPH PROGRAM FOR PROMISS

```
data a;
INFILE 'RAND1.CPR' FIRSTOBS=2;input x y;
GOPTIONS DEVICE=HP7470;
proc gplot;
   axis1 label=(h=1 f=simplex 'LOG OF CYCLES')
         value=(h=1 f=simplex);
   axis2 value=(h=1 f=simplex) label=none;
   plot y*x / haxis=axis1 vaxis=axis2;
   TITLE H=1 A=90 F=SIMPLEX 'PROBABILITY DENSITY FUNCTION';
   symbol i=spline v=square;
data B;
INFILE 'RAND2.CPR' FIRSTOBS=2;input x y;
proc gplot;
   axis1 label=(h=1 f=simplex 'LOG OF CYCLES')
         value=(h=1 f=simplex);
   axis2 value=(h=1 f=simplex) label=none;
   plot y*x / haxis=axis1 vaxis=axis2;
   TITLE H=1 A=90 F=SIMPLEX 'CUMULATIVE DISTRIBUTION FUNCTION';
   symbol i=spline v=square;
```